

A Knowledge Acquisition Method for Fuzzy Expert Systems – An Improvement via Mountain-Clustering

Antonio C.S. Branco^{*}, Heloisa Camargo^{**}, Alexandre Evsukoff^{***}, Doris F. de Aragon^{*} & Luiz F. Pereira^{*}

^{*}*ILTC, Instituto de Lógica, Filosofia e Teoria da Ciência
Rua Almirante Teffé 637, 24030-080 – Niterói – Rio
de Janeiro- Brazil*

branco@iltc.uff.br, doris@iltc.uff.br, lfsmp@iltc.uff.br

^{**}*UFSCar, Universidade Federal de São Carlos
Via Washington Luis, km 235 – C.P. 676, 13565-905 – São
Carlos – SP - Brazil*

heloisa@leao.dcc.ufscar.br

^{***}*CNRS-INPG-UJF, Laboratoire d'Automatique de
Grenoble, BP 46, 38402 Saint Martin d'Hères, France
evsukoff@lag.ensieg.inpg.fr*

Abstract

This work presents a knowledge acquisition method for fuzzy systems applied to pattern recognition problems. An improvement to an earlier version of this method (Evsukoff[3]) is proposed through the use of a clustering technique to generate the fuzzy partitions of the input space. An adaptation of the Mountain-Clustering method (Yager[12]) was applied and two different approaches of clustering for generate the input fuzzy partitions have been developed. These clustering approaches were used with the supervised learning algorithm and were tested upon some classification problems found in the literature.

1 Introduction

The tasks of classification and pattern recognition have been widely investigated and there are many works describing the most popular techniques applied to this area (Blekas[1], Ishibuchi[5], Jang[6], Joshi[7], Nauck[8], Quinlan[10], Ripley[11] and others). The problem of learning from examples is present in all of them. This process can be done in a supervised or unsupervised way.

acquisition that generates a knowledge base in the form of zero-order Sugeno rules from input data through a supervised learning algorithm. The method (devoted to diagnosis problems) derived from the work developed in Branco[2], where a heuristic knowledge acquisition from domain experts was proposed. The method presented in Evsukoff[3] is an extension of that proposed in Branco[2], where the uncertainty propagation problem is solved by a decomposition schema. A unified representation allows the knowledge to be elicited from experts or learned from a database. The major drawback of that version of the method was its dependency on the number of fuzzy sets in each fuzzy partition and their parameterization.

Further research has been done to improve the classification performance of the method (Evsukoff[4]). The present work introduces an extension of the method described in Evsukoff[3]: the fuzzy partitions in input space are computed automatically by clustering, instead of being defined by trial and error. A modification of the Mountain-Clustering Method is proposed in two different approaches to the construction of the input fuzzy partition. The first approach applies the clustering method to each feature with respect to all classes at the same time. In the second one, each pair feature-class is considered independently. The membership functions generated by the clustering method are triangular and the partitions are always normalized and strong.

The paper is organized as follows: section 2 presents a description of the knowledge acquisition method introduced in Evsukoff[3]; in section 3, the Mountain-Clustering Method is briefly described; in section 4, the two approaches to the application of the data clustering concept investigated in this work are presented, as well as the modifications to the Mountain-Clustering Method adopted in the analysis; section 5 presents the results of the application of the learning algorithm in some classification problems found in the literature; finally, section 6 presents the conclusions and future work.

2 The Knowledge Acquisition Method

2.1 The Inference Mechanism

Many diagnosis problems can be seen as a pattern recognition problem, when the diagnostics can be concluded from the values of some features. The diagnostics are considered as classes. The set of all classes is noted by $C: \{C_1, \dots, C_K\}$. Each feature is denoted by a fuzzy partition. The set of

all features is noted by $X=\{X_1...X_n\}$, where $X_i=(X_i, \mu_{X_i}(x_i))$ is the i^{th} feature fuzzy partition of n_i fuzzy sets in the universe U_i of the values x_i .

For convenience, strong normalised fuzzy partitions are used. A fuzzy partition $A=(A, \mu_{A_i}(\xi))$, of n fuzzy sets in the universe Ω is said to be normalised and strong if:

$$\begin{aligned} &(\forall i)(\exists \xi_0 \in \Omega)(\mu_{A_i}(\xi_0) = 1) \\ &(\forall \xi \in \Omega)(\sum_i \mu_{A_i}(\xi) = 1). \end{aligned} \quad (1)$$

For given feature values $\mathbf{x}^0=\{x_1^0...x_n^0\}$, the objective of the proposed method is to construct a zero-order Sugeno rule base which can correctly order the set of classes by their membership values.

The knowledge acquisition process (Branco[2]) consists in obtain, for each feature X_i , a table of influences (Table 1) where lines are associated to classes and columns are associated to fuzzy sets in the feature fuzzy partition.

Each element $\phi_{kj}^i \in [0,1]$ in this table represents the influence degree of the feature fuzzy set X_{ij} upon the occurrence of class C_k . We will sometimes refer to this values as the matrix Φ^i . When expert knowledge is available, the influence values can be elicited from an expert. Otherwise, when a data set is available, they can be obtained by a supervised learning algorithm. This work focuses on the supervised learning algorithm way.

Table 1: Table of influences

	X_i				
	X_{i1}	...	X_{ij}	...	X_{in_i}
C_1	ϕ_{11}^i	...	ϕ_{1j}^i	...	$\phi_{1n_i}^i$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
C_k	ϕ_{k1}^i	...	ϕ_{kj}^i	...	$\phi_{kn_i}^i$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
C_K	ϕ_{K1}^i	...	ϕ_{Kj}^i	...	$\phi_{Kn_i}^i$

From each feature table, a zero-order Sugeno rule base can be written as:

$$\text{If } x_i \text{ is } X_{ij} \text{ then } \pi_{kj}^i = \phi_{kj}^i, \quad k = 1..K \quad (2)$$

where π_{kj}^i is the output of the j^{th} rule relative to the i^{th} feature rule base associated to the class C_k , given the input value x_i^0 . It can be shown that

π_k^i is the membership of the input x_i^0 to the class C_k computed from the i^{th} feature.

If strong and normalized fuzzy partitions are used, the output π_k^i of the rule base represented by eqn(2) is computed as:

$$\pi_k^i = \sum_{j=1..n_i} \phi_{kj}^i w_{ij} \quad (3)$$

where $w_{ij} = \mu_{x_{ij}}(x_i^0)$ is the fire strength of each rule eqn(2).

For an input $\mathbf{x}^0 = \{x_1^0 \dots x_N^0\}$, the final output π_k for the class C_k in case of conjunctive aggregation of the N features rule bases, can be computed by:

$$\pi_k = \prod_{i=1..N} \pi_k^i. \quad (4)$$

This final output π_k after the aggregation of all features is the membership of the input $\mathbf{x}^0$ to the class C_k .

The final decision can be taken from the class membership by a decision rule, as is usual in decision problems (Zimmermann[13]). The simplest rule is the maximum rule which consists in choosing the class with the greatest output value, as used in this work.

2.2 The Learning Algorithm

Consider now a training set composed by M correctly classified examples. Each instance in the training set is represented by a couple $(\mathbf{x}^m, \mathbf{y}^m)$, where $\mathbf{x}^m = \{x_1^m \dots x_N^m\}$ are the values of the N features and $\mathbf{y}^m = \{y_1^m \dots y_K^m\}$ are the membership values of $\mathbf{x}^m$ to each class.

For each feature, its fuzzy partition (with n_i fuzzy sets) will define a point in I^{n_i} (the hypercube of dimension n_i) for each instance of the training set. The points associated to the same class C_k will form a cluster in I^{n_i} . If the examples represent correctly the distribution of class C_k in the universe U_i , such a cluster can be associated to the class C_k .

The co-ordinates of the center of gravity of a cluster associated to a class C_k represent the frequency with which the feature fuzzy sets influence the class C_k . Thus, it is proposed to use such co-ordinates as the point $\phi_k = \{\phi_{k1}^i \dots \phi_{kn}^i\}$ which is the k^{th} line in the matrix Φ^i .

Each co-ordinate of the center of gravity of a cluster in I^{n_i} associated to a class C_k can be computed as the average of the co-ordinates of all points in the cluster, weighted by the membership value of the class:

$$\phi_{kj}^i = \frac{\sum_{m=1..M} w_{ij}^m y_k^m}{\sum_{m=1..M} y_k^m} \quad (5)$$

A supervised learning algorithm was implemented from eqn(5) to obtain the matrix given by Table 1.

3 The Mountain-Clustering Method

The Mountain-Clustering Method is a three step method: first, a discretization of the space domain of the training data set is done to generate the potential cluster centers; then, the mountain function is constructed and finally, by an iterative process of destruction of the mountain function, the cluster centers are generated.

Consider the training set of M points as described above. For each feature X_i , the data to be clustered is the collection of points $O_m(x_i^m, y_k^m)$, $m=1..M$, in the space $X_i \times Y$, where x_i^m is the input data (feature value) and y_k^m is the output data (membership of the x^m to the class C_k). The first step in the mountain method is to form a discretization of the $X_i \times Y$ space by griding X_i and Y with $r_1 \in r_2$ lines, respectively. The lines does not have to be necessarily equidistant. The intersections of these lines form the set of potential clusters, denoted by P . An element in P is denoted as $P_{cl}(x_c, y_l)$ to indicate the point obtained by the intersection of the grid lines passing through x_c and y_l .

The second step is the construction of the mountain function, $F:P \rightarrow \mathfrak{R}$, which is defined on the set P of potential cluster centers by:

$$F(P_{cl}) = \sum_{m=1..M} e^{-(\alpha d(P_{cl}, O_m))} \quad (6)$$

where $\mathfrak{R}$ is the set of real numbers, O_m is the m^{th} data point, (x_i^m, y_k^m) , α is a positive constant, and $d(P_{cl}, O_m)$ is a measure of distance between P_{cl} and O_m , typically the Euclidean distance.

In the third step, we generate the cluster centers by an iterative process of revision of the mountain function. Starting with the revised mountain function F_p (consider the mountain function F as the revised mountain function F_l), the process is as follows:

- Find $F_p^* = \text{Max}_{cl} [F_p(P_{cl})]$.
- Designate p^{th} cluster center at P_p^* , the location of maximal node found in a).
- Formulate revised mountain function, F_{p+1} , as

$$F_{p+1}(P_{cl}) = F_p(P_{cl}) - F_p^* e^{-\beta d(P_p^*, P_{cl})} \quad (7)$$

where β is a positive constant and the others symbols denote the same as before.

The stop criterion to this process can be at step s with the estimation of s cluster centers after next mountain peak F_{s+1}^* becomes less than a given stopping constant δ :

$$F_{s+1}^* < \delta \quad (8)$$

The Mountain-Clustering Method was initially proposed as a procedure to obtain the structure and initial estimates of the parameters of the rule-base of a fuzzy system model, as described in Yager[12]. In this work we just applied it to construct the fuzzy partitions of the features (input) domains, as described in the next section.

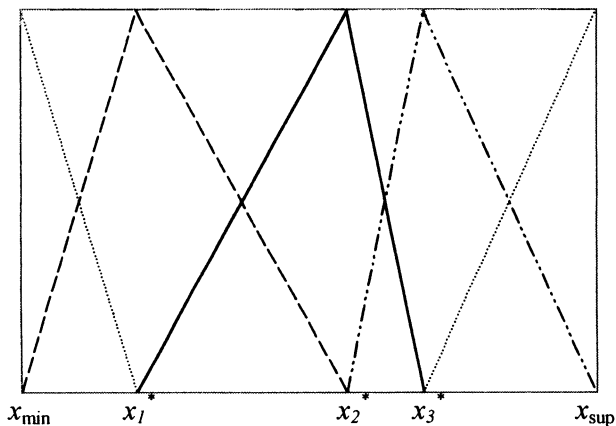
4 The Application of the Clustering Technique

The major drawback of the method described in Evsukoff[3], was the generation of the fuzzy partitions of the non-Boolean features domains. In Evsukoff[3], the partitions were constructed by fuzzy sets defined through triangular membership functions equally distributed over the features domains. The number of fuzzy sets in the partition was the same for all features and defined empirically by a trial and error process. These restrictions to fuzzy partitions are of pragmatic nature and are not inherent to the proposed method. They were assumed as a means to generate the partition using a computationally simple process, as long as the main focus of the work was the learning method itself.

In the present work, it is proposed and used an adaptation of the Mountain-Clustering Method described above to generate the fuzzy partitions in the input space. In Yager[12], the membership functions of the fuzzy sets in the rules antecedents are assumed to be of Gaussian-type. In this work it is assumed that the fuzzy sets are defined by triangular membership functions and that the fuzzy partitions are always normalized and strong.

Once the Mountain-Clustering Method has determined the cluster centers, the projections of these centers on the input space are considered as the centers of the triangular membership functions of the fuzzy sets in the partition. Independently of the cluster centers found, we always considered two fuzzy sets of the partition centered at the minimum and at the supreme of the input domain. As an example, in Figure 1 it is shown the partition constructed from three cluster centers (determined by the Mountain-Clustering Method), whose projections on the input domain are

Figure 1: Fuzzy partition generated by 3 cluster centers



The use of triangular functions instead of Gaussian functions required an adaptation of the mountain function used in the method. In this work we used the mountain function defined as:

$$F(P_{cl}) = \sum_{m=1..M} (1 - \alpha (d(P_{cl}, O_m))) \quad (9)$$

where $d(P_{cl}, O_m) = |x_c - x_i^m|$ is the absolute value of the difference $x_c - x_i^m$ and the others symbols denote the same elements as before.

In the third step of the method, - the destroying of the mountain function – the mountain function is revised as

$$F_{p+1}(P_{cl}) = F_p(P_{cl}) - F_p^* \times (1 - \beta (d(P_p^*, P_{cl}))) \quad (10)$$

where F_p^* is the value of the mountain function corresponding to cluster P_p^* just identified and the others symbols denote the same elements as above.

In the following, two different approaches to the construction of a classification system under this proposal are analyzed.

4.1 A Fuzzy Partition for each Feature

The Mountain-Clustering Method, with the modifications presented, is used to generate the fuzzy partition of the domain of each non-Boolean feature X_i , considering as the data to be clustered the set of points $O_m(x_i^m, y_k^m)$, $m=1..M$, $k=1..K$, $y_k^m \neq 0$.

In this approach, the fuzzy partition of each feature is the same for all classes, what allows keeping all the characteristics of the method defined in Evsukoff[3].

4.2 A Fuzzy Partition for each pair Feature-Class

The Mountain-Clustering Method, with the modifications presented, is used to generate a fuzzy partition of the domain of each non-Boolean feature, considering each class separately. Thus, for each non-Boolean feature considered as an input variable, each class is viewed separately as an output variable. For each feature-class pair, X_i-C_k , the clustering method is applied to the set of points $O_m(x_i^m, y_k^m)$, $m=1..M$, $y_k^m \neq 0$, generating a different partition for each one.

So, in this approach, for the same feature X_i , we have a different partition of its domain for each class. This implies in a reformulation of Table 1. Now, for each pair X_i-C_k , we have something like Table 2, where the symbols denote the same as in Table 1:

Table 2: Influence degrees of feature X_i over class C_k .

	X_i				
	X_{i1}^k	...	X_{ij}^k	...	$X_{in_i}^k$
C_k	ϕ_{k1}^i	...	ϕ_{kj}^i	...	$\phi_{kn_i}^i$

From each feature table, a zero-order Sugeno rule base can be written as:

$$\text{if } x_i \text{ is } X_{ij}^k \text{ then } \pi_{kj}^i = \phi_{kj}^i, j=1..n_i^k \quad (11)$$

where π_{kj}^i is the output of the j^{th} rule relative to the i^{th} feature rule base associated to the class C_k , given the input value x_i^0 and X_{ij}^k is the fuzzy set associated to the fuzzy partition of X_i computed for the class C_k .

The output π_k^i of the rule base represented by eqn(11) is computed as in eqn(3) and the final output π_k for the class C_k , in case of conjunctive aggregation of the N features rule bases, for an input $\mathbf{x}^0 = \{x_1^0 \dots x_N^0\}$, is still computed by eqn(4).

5 Results

These two described approaches for clustering were applied to some of the problems of the benchmark PROBEN1 (Prechelt[9]) tested in Evsukoff[3]. The minimal error rates obtained using the proposed approaches are given in Table 3 in the second and third columns corresponding to the descriptions in sections 4.1 and 4.2, respectively. The fourth column presents the results obtained in Evsukoff[3]. The fifth column presents the results of a multi-layer neural network training (Prechelt[9]). The sixth column shows some results of the machine learning algorithm C4.5 (Quinlan[10]), applied to the same data sets but not in the format found in PROBEN1.

Table 3: Results on classification examples

	Method error rate (%)				
Example	4.1	4.2	Ev97	ANN	C4.5
cancer1	0.57	0.57	0.57	1.38	5.69
cancer2	2.29	2.87	3.45	4.77	
cancer3	2.29	2.29	1.72	3.70	
card1	12.79	12.21	19.65	14.05	13.6
card2	14.53	13.37	12.72	18.91	
card3	15.11	14.53	17.34	18.84	
diabetes1	22.40	21.87	22.92	24.10	26.9
diabetes2	18.23	18.23	20.31	26.42	
diabetes3	21.35	19.79	21.88	22.59	
glass1	39.62	39.62	35.85	32.70	37.0
glass2	37.73	37.73	58.49	55.57	
glass3	43.40	37.73	43.40	58.40	
heartc1	14.66	14.66	20.00	20.82	22.6
heartc2	6.67	5.33	10.67	5.13	
heartc3	13.33	12.00	13.33	15.40	
horse1	26.37	26.37	32.97	29.19	-
horse2	34.07	34.07	34.07	35.86	
horse3	29.67	29.67	35.16	34.16	

6 Conclusions

In this paper, we proposed two approaches to construct the fuzzy partitions of the input space in a fuzzy classification system using an adaptation of the Mountain-Clustering Method.

The proposed approaches improved the performance of an earlier version of the method, as can be observed in Table 3, especially in the problems card1, glass2, heartc1, heartc2, horse1 and horse3. The approach described in section 4.2 has achieved better results than that of section 4.1, but only in a few problems.

Further work will be made in order to use these approaches in conjunction with that presented in Evsukoff[4] and others membership functions will be tested to construct the fuzzy partitions.

Acknowledgements

Antonio C.S. Branco, Alexandre Evsukoff, Doris F. de Aragon and Luiz F. Pereira are supported by the Brazilian National Research Council (CNPq) under the grants 142510/98-0, 200482/95-5, 680069/95-0 and 680069/95-0, respectively.

References

- [1] Blekas, K.; Likas, A.; Stafylopatis, A. – A Fuzzy Neural Network Approach to Classification Based on Proximity Characteristics of Patterns – Proceedings of 9th IEEE International Conference on Tools with Artificial Intelligence, ICTAI'97, California, USA, 1997.
- [2] Branco, A.C.S.; Silva, L.F.; Evsukoff, A. and Aragon, D.: An Expert system applied to the petroleum industry - II World Congress on Expert Systems, Lisbon, 1994.
- [3] Evsukoff, A.; Branco, A.C.S.; Gentil, S. – A Knowledge Acquisition Method for Fuzzy Expert Systems in Diagnosis Problems – Sixth IEEE International Conference on Fuzzy Systems – FUZZ-IEEE'97, 1997.
- [4] Evsukoff, A.; Gentil, S. – Symbolic-numeric Integration in Pattern Recognition – European Workshop of Fuzzy Decision Analysis and Neural Networks – EFDAN'98, 1998.
- [5] Ishibuchi, H. ; Morisawa, T. ; Nakashima, T. – Voting Schemes for Fuzzy-Rule-Based Classification Systems – Proceedings of 5th FUZZ-IEEE, 614-620, 1996.



- [6] Jang, J.-S. R.; Sun, C.-T. – Neuro-Fuzzy Modeling and Control – Proceedings of the IEEE, vol.83, no.3, 1995.
- [7] Joshi, A.; Ramakrishnan, N.; Houstis, E.N.; Rice, J.R. – On Neurobiological, Neuro-Fuzzy, Machine Learning and Statistical Pattern Recognition Techniques – IEEE Transactions on Neural Networks, 8, no.1, 1997.
- [8] Nauck, D.; Klawonn, F. – Neuro-Fuzzy Classification Initialized by Fuzzy Clustering – Proceedings of EUFIT'96, 1996.
- [9] Prechelt, L.: *PROBEN1*-A set of neural network benchmark problems and benchmarking rules - Universität Karlsruhe Technical Report 21/94, 1994.
- [10] Quinlan, J. R.: Improved use of continuous attributes in C4.5 - Journal of Artificial Intelligence Research 4, pp. 77-90, 1996.
- [11] Ripley, B.D. – Pattern Recognition and Neural Networks – Cambridge University Press, 1996.
- [12] Yager, R.R.; Filev, D. – Essentials of Fuzzy Modeling and Control – John Wiley and Sons, Inc., 1994.
- [13] Zimmermann, H.-J. – Fuzzy Set Theory and its Applications – Kluwer, 1996.